

MMM	MMM	TTTTTTTTTTTTTTTT	AAAAAAAAA	AAAAAAAAA	CCCCCCCCCCCC	PPPPPPPPPPPP	
MMM	MMM	TTTTTTTTTTTTTTTT	AAAAAAAAA	AAAAAAAAA	CCCCCCCCCCCC	PPPPPPPPPPPP	
MMM	MMM	TTTTTTTTTTTTTTTT	AAAAAAAAA	AAAAAAAAA	CCCCCCCCCCCC	PPPPPPPPPPPP	
MMMMMM	MMMMMM	TTT	AAA	AAA	CCC	PPP	PPP
MMMMMM	MMMMMM	TTT	AAA	AAA	CCC	PPP	PPP
MMMMMM	MMMMMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPP	PPP
MMM	MMM	TTT	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPPPPPPPPPPP	
MMM	MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC	PPP	
MMM	MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC	PPP	
MMM	MMM	TTT	AAAAAAAAAAAAAAAA	AAAAAAAAAAAAAAAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCC	PPP	
MM/	MMM	TTT	AAA	AAA	CCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCCCCCCCCCCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCCCCCCCCCCC	PPP	
MMM	MMM	TTT	AAA	AAA	CCCCCCCCCCCC	PPP	

```
FFFFFFFFF  RRRRRRRR  MM      MM  HH      HH  DDDDDDDD  RRRRRRRR
FFFFFFFFF  RRRRRRRR  MM      MM  HH      HH  DDDDDDDD  RRRRRRRR
FF          RR      RR  MMMM  MMMM  HH      HH  DD      DD  RR      RR
FF          RR      RR  MMMM  MMMM  HH      HH  DD      DD  RR      RR
FF          RR      RR  MM  MM  MM  HH      HH  DD      DD  RR      RR
FFFFFFFFF  RRRRRRRR  MM      MM  HHHHHHHHHH  DD      DD  RRRRRRRR
FFFFFFFFF  RRRRRRRR  MM      MM  HHHHHHHHHH  DD      DD  RRRRRRRR
FF          RR      RR  MM      MM  HH      HH  DD      DD  RR      RR
FF          RR      RR  MM      MM  HH      HH  DD      DD  RR      RR
FF          RR      RR  MM      MM  HH      HH  DD      DD  RR      RR
FF          RR      RR  MM      MM  HH      HH  DD      DD  RR      RR
FF          RR      RR  MM      MM  HH      HH  DDDDDDDD  RR      RR
FF          RR      RR  MM      MM  HH      HH  DDDDDDDD  RR      RR
```

```
LL          IIIIII  SSSSSSSS
LL          IIIIII  SSSSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SSSSSS
LL          II      SSSSSS
LL          II      SS
LL          II      SS
LL          II      SS
LL          II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

```
1 0001 0 MODULE FRMHDR (LANGUAGE (BLISS32) ,
2 0002 0 IDENT = 'V04-000' ,
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1
31 0031 1 FACILITY: MTAACP
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module formats HDR1, HDR2, HDR3, and HDR4.
36 0036 1
37 0037 1 ENVIRONMENT:
38 0038 1
39 0039 1 Starlet operating system, including privileged system services
40 0040 1 and internal exec routines.
41 0041 1
42 0042 1 --
43 0043 1
44 0044 1
45 0045 1
46 0046 1 AUTHOR: D. H. GILLESPIE, CREATION DATE: 2-JUN-77 14:35
47 0047 1
48 0048 1 MODIFIED BY:
49 0049 1
50 0050 1 V03-003 MMD0301 Meg Dumont, 20-Jun-1984 11:26
51 0051 1 Fix to default HDR4 file extension to ASCII zeros instead of
52 0052 1 decimal zeros
53 0053 1
54 0054 1 V03-002 MMD0279 Meg Dumont, 23-Mar-1984 10:25
55 0055 1 Fix long file name support such that for ANSI version
56 0056 1 3 volumes it converts the exentsion length to
57 0057 1 ASCII characters before writing it to the label.
```



```
58 0058 1
59 0059 1
60 0060 1
61 0061 1
62 0062 1
63 0063 1
64 0064 1
65 0065 1
66 0066 1
67 0067 1
68 0068 1
69 0069 1
70 0070 1
71 0071 1
72 0072 1
73 0073 1
74 0074 1
75 0075 1
76 0076 1
77 0077 1
78 0078 1
79 0079 1
80 0080 1
81 0081 1
82 0082 1
83 0083 1
84 0084 1
85 0085 1
86 0086 1
87 0087 1
88 0088 1
89 0089 1
90 0090 1
91 0091 1
92 0092 1
93 0093 1
94 0094 1
95 0095 1
96 0096 1
97 0097 1
98 0098 1
99 0099 1
100 0100 1
101 0101 1
102 0102 1
103 0103 1
104 0487 1
105 0488 1
106 0489 1
```

V03-001 MMD0160 Meg Dumont, 26-Apr-1983 9:31
Add long file name support include: 1) Change FORMAT_FILE_NAME
to understand that VMS long file names are split between
the HDR1 and HDR4 labels. 2) Change FORMAT_HDRS to format the
HDR4. Added support for interchange qualifier.

V02-012 DMW00069 David Michael Walp 11-Jan-1981
Added support of ANSI "a" 17 character filename thru
QIO filename parameter

V02-011 DMW00064 David Michael Walp 6-Jan-1981
Return VMS file spec created by ASCNAME without quotes

V02-010 DMW00063 David Michael Walp 18-Dec-1981
Finished ANSI "a" 17 character file name support

V02-009 DMW00053 David Michael Walp 10-Nov-1981
Return if ANSI resultant name, return it minus trailing
spaces.

V02-008 DMW00043 David Michael Walp 27-Oct-1981
Added ANSI "a" 17 character file name support

V02-007 DMW00016 David Michael Walp 20-May-1981
Get the File Set Id from the MVL rather than 1st volume
mounted label in the VCB.

V02-006 DMW00008 David Michael Walp 23-Jan-1981
Added check for "X" wild card, needed beacuse of expanded
wild card support. Also code commented out for support of
HDR2 attributes.

V02-005 REFORMAT Maria del C. Nasr 30-Jun-1980

A0004 MCN0008 Maria del C. Nasr 22-Feb-1980 16:29
Temporary support of RMS attributes in HDR2

A0003 MCN0003 Maria del C. Nasr 28-Sep-79 10:23
Add HDR3 processing

LIBRARY 'SYS\$LIBRARY:LIB.L32';

REQUIRE 'SRC\$:MTADEF.B32';

FORWARD ROUTINE
FORMAT_HDRS : COMMON_CALL NOVALUE: ! format headers

```
108 M 0490 1 MACRO RAD50_TO_VMS ( STRING, VERSION, OUT_INDEX ) =
109 M 0491 1
110 M 0492 1 ++
111 M 0493 1
112 M 0494 1 FUNCTIONAL DESCRIPTION:
113 M 0495 1 This routine converts a RAD-50 file name block into the
114 M 0496 1 equivalent VMS format name. Long file names are not supported
115 M 0497 1 in RAD50 mode.
116 M 0498 1
117 M 0499 1 CALLING SEQUENCE:
118 M 0500 1 RAD50_TO_VMS ( ARG1, ARG2, ARG3 )
119 M 0501 1
120 M 0502 1 INPUT PARAMETERS:
121 M 0503 1 none
122 M 0504 1
123 M 0505 1 IMPLICIT INPUTS:
124 M 0506 1 NMBLOCK - the Radix 50 name block
125 M 0507 1
126 M 0508 1 OUTPUT PARAMETERS:
127 M 0509 1 ARG1 - buffer for file name string
128 M 0510 1 ARG2 - word to receive version number
129 M 0511 1 ARG2 - size of filename string
130 M 0512 1
131 M 0513 1 IMPLICIT OUTPUTS:
132 M 0514 1 none
133 M 0515 1
134 M 0516 1 SIDE EFFECTS:
135 M 0517 1 none
136 M 0518 1
137 M 0519 1 --
138 M 0520 1
139 M 0521 1 BEGIN
140 M 0522 1
141 M 0523 1 EXTERNAL
142 M 0524 1 NMBLOCK : VECTOR [ , WORD ]; ! the rad50 name block
143 M 0525 1
144 M 0526 1 MAP
145 M 0527 1 STRING : VECTOR [ , BYTE ], ! string buffer arg
146 M 0528 1 VERSION : WORD, ! version number arg
147 M 0529 1 OUT_INDEX : LONG; ! file name size arg
148 M 0530 1
149 M 0531 1 LOCAL
150 M 0532 1 CHARS : VECTOR [ 3, BYTE ]; ! holding place for characters
151 M 0533 1
152 M 0534 1
153 M 0535 1 ! Set up the index. Then start up the outer loop, which iterates
154 M 0536 1 ! over name and type fields.
155 M 0537 1
156 M 0538 1 OUT_INDEX = 0;
157 M 0539 1
158 M 0540 1 INCR K FROM 0 TO 3 BY 3 DO
159 M 0541 1 BEGIN
160 M 0542 1
161 M 0543 1 ! The next loop iterates over the RAD-50 words in the name block.
162 M 0544 1 ! There are 3 words for name, 1 for type. Expand each word into
163 M 0545 1 ! the 3 RAD-50 characters.
164 M 0546 1
```

```

: 165      M 0547 1
: 166      M 0548 1
: 167      M 0549 1
: 168      M 0550 1
: 169      M 0551 1
: 170      M 0552 1
: 171      M 0553 1
: 172      M 0554 1
: 173      M 0555 1
: 174      M 0556 1
: 175      M 0557 1
: 176      M 0558 1
: 177      M 0559 1
: 178      M 0560 1
: 179      M 0561 1
: 180      M 0562 1
: 181      M 0563 1
: 182      M 0564 1
: 183      M 0565 1
: 184      M 0566 1
: 185      M 0567 1
: 186      M 0568 1
: 187      M 0569 1
: 188      M 0570 1
: 189      M 0571 1
: 190      M 0572 1
: 191      M 0573 1
: 192      M 0574 1
: 193      M 0575 1
: 194      M 0576 1
: 195      M 0577 1
: 196      M 0578 1
: 197      M 0579 1
: 198      M 0580 1
: 199      M 0581 1
: 200      M 0582 1
: 201      M 0583 1
: 202      M 0584 1
: 203      M 0585 1

      INCR I FROM 0 TO ( IF .K THEN 0 ELSE 2 )
      DO
      BEGIN
      CHARS [ 0 ] = .NMBLOCK [ .I + .K ] / ( 40 * 40 );
      CHARS [ 1 ] = ( .NMBLOCK [ .I + .K ] / 40 ) MOD 40;
      CHARS [ 2 ] = .NMBLOCK [ .I + .K ] MOD 40;

      ! Now convert each character into the correct ASCII code and store
      ! it in the string buffer if it is not null.
      INCR J FROM 0 TO 2 DO
      IF .CHARS [ .J ] NEQ 0 THEN
      BEGIN
      STRING [ .OUT_INDEX ] = ( IF .CHARS [ .J ] LSS 30
      THEN ( .CHARS [ .J ] - 1 ) + 'A'
      ELSE ( .CHARS [ .J ] - 30 ) + '0' );

      OUT_INDEX = .OUT_INDEX + 1;
      END;

      END; ! end of word loop

      ! At the end of name field, insert the dot.
      IF .K EQL 0
      THEN
      BEGIN
      STRING [ .OUT_INDEX ] = '.';
      OUT_INDEX = .OUT_INDEX + 1;
      END;
      END; ! end of outer loop

      ! fill in the version number
      VERSION = .NMBLOCK [ 4 ];

      END;
      X; ! end of macro RAD50_TO_ASCII
```



```

205 M 0586 1 MACRO FORMAT_FILE_NAME =
206 M 0587 1
207 M 0588 1 |++
208 M 0589 1
209 M 0590 1 FUNCTIONAL DESCRIPTION:
210 M 0591 1 formats the File's Name, Type and Version to placed into the header
211 M 0592 1
212 M 0593 1 CALLING SEQUENCE:
213 M 0594 1 FORMAT_FILE_NAME
214 M 0595 1
215 M 0596 1 INPUT PARAMETERS:
216 M 0597 1 none
217 M 0598 1
218 M 0599 1 IMPLICIT INPUTS:
219 M 0600 1 none
220 M 0601 1
221 M 0602 1 OUTPUT PARAMETERS:
222 M 0603 1 none
223 M 0604 1
224 M 0605 1 IMPLICIT OUTPUTS:
225 M 0606 1 file name is formatted in the HDR1 and HDR4 labels
226 M 0607 1
227 M 0608 1 SIDE EFFECTS:
228 M 0609 1 none
229 M 0610 1
230 M 0611 1 |--
231 M 0612 1
232 M 0613 1 BEGIN
233 M 0614 1
234 M 0615 1 EXTERNAL ROUTINE
235 M 0616 1 CALC TAPE VER, ! turn VMS version number
236 M 0617 1 PARSE_NAME TYPE, ! into ANSI generation nums
237 M 0618 1 PARSE-QUOTED NAME: COMMON CALL NOVALUE, ! parse file name string
238 M 0619 1 RESULTANT STRING: COMMON CALL NOVALUE, ! parse a spec in quotes
239 M 0620 1 STRIP VERSION : COMMON CALL, ! return resultant string
240 M 0621 1 SYS$F$O : ADDRESSING_MODE ( ABSOLUTE ); ! strip version from file spec
241 M 0622 1 ! format generation num
242 M 0623 1
243 M 0624 1 EXTERNAL
244 M 0625 1 ANSI_NAME_SIZE : SIGNED BYTE, ! size of the ANSI file name
245 M 0626 1 HDR1 : REF BBLOCK, ! point to the HDR1
246 M 0627 1 HDR4 : REF BBLOCK, ! point to the HDR4
247 M 0628 1 IO_PACKET : REF BBLOCK, ! pointer to current IRP
248 M 0629 1 LOCAL_FIB : BBLOCK; ! copy of users File Info Blk
249 M 0630 1
250 M 0631 1 LOCAL
251 M 0632 1 ABD : REF BBLOCKVECTOR [ , ABD$C_LENGTH ],
252 M 0633 1 ! pointer to ACP buffer desc
253 M 0634 1 DESCRPT : VECTOR [ 2, LONG ], ! general purpose descriptor
254 M 0635 1 FILE_SPEC_PTR : LONG, ! point to file name buffer
255 M 0636 1 FILE_SPEC_LEN : LONG, ! length of file name buffer
256 M 0637 1 NAME_STRING : VECTOR [ FILE_SPEC_MAX, BYTE ],
257 M 0638 1 ! buff to hold converted RAD50
258 M 0639 1 FILE_ID : VECTOR [ FILE_SPEC_MAX, BYTE ], ! FILE ID
259 M 0640 1 GEN_NUM_VER : VECTOR [ 2, LONG ], ! ANSI version numbers
260 M 0641 1 QUOTED_NAME : BITVECTOR [ 1 ], ! was the spec passed in quotes
261 M 0642 1 VERSION : WORD; ! VMS version number

```

```

: 262      M 0643 1
: 263      M 0644 1
: 264      M 0645 1
: 265      M 0646 1
: 266      M 0647 1
: 267      M 0648 1
: 268      M 0649 1
: 269      M 0650 1
: 270      M 0651 1
: 271      M 0652 1
: 272      M 0653 1
: 273      M 0654 1
: 274      M 0655 1
: 275      M 0656 1
: 276      M 0657 1
: 277      M 0658 1
: 278      M 0659 1
: 279      M 0660 1
: 280      M 0661 1
: 281      M 0662 1
: 282      M 0663 1
: 283      M 0664 1
: 284      M 0665 1
: 285      M 0666 1
: 286      M 0667 1
: 287      M 0668 1
: 288      M 0669 1
: 289      M 0670 1
: 290      M 0671 1
: 291      M 0672 1
: 292      M 0673 1
: 293      M 0674 1
: 294      M 0675 1
: 295      M 0676 1
: 296      M 0677 1
: 297      M 0678 1
: 298      M 0679 1
: 299      M 0680 1
: 300      M 0681 1
: 301      M 0682 1
: 302      M 0683 1
: 303      M 0684 1
: 304      M 0685 1
: 305      M 0686 1
: 306      M 0687 1
: 307      M 0688 1
: 308      M 0689 1
: 309      M 0690 1
: 310      M 0691 1
: 311      M 0692 1
: 312      M 0693 1
: 313      M 0694 1
: 314      M 0695 1
: 315      M 0696 1
: 316      M 0697 1
: 317      M 0698 1
: 318      M 0699 1

! which filename should be used
! get the filename from name block if not specified as attribute
ABD = .BBLOCK [ .IO PACKET [ IRPSL SVAPTE ], AIB$ DESCRPT ];
IF .ABD [ ABD$C_NAME, ABD$W_COUNT ] EQLU 0
THEN
    BEGIN
        RAD50 TO VMS ( NAME STRING, VERSION, FILE_SPEC_LEN );
        FILE_SPEC_PTR = NAME_STRING;
        QUOTED_NAME [ 0 ] = FALSE;
    END
ELSE
    BEGIN
        FILE_SPEC_LEN = .ABD [ ABD$C_NAME, ABD$W_COUNT ];
        FILE_SPEC_PTR = .ABD [ ABD$C_NAME, ABD$W_TEXT ] +
            ABD [ ABD$C_NAME, ABD$W_TEXT ] + 1;

        ! do not allow wild cards in the version field
        !
        VERSION = STRIP_VERSION ( FILE_SPEC_LEN,
            FILE_SPEC_PTR,
            FALSE,
            QUOTED_NAME [ 0 ] );

    END;

! check that it is not too large
!
IF .VERSION GTRU 32767 THEN ERR_EXIT ( SS$_BADFILEVER );

! Space fill the temporary FILE_ID field
CH$FILL(' ', FILE_SPEC_MAX, FILE_ID);

! parse the file name if it is a VMS file spec and place into HDR1
!
IF .ANSI_NAME_SIZE LSS 0
THEN
    BEGIN
        DESCRPT [ 0 ] = FILE_SPEC_MAX;
        DESCRPT [ 1 ] = FILE_ID[0];

        ! call the correct parse routine
        !
        IF .QUOTED_NAME [ 0 ]
        THEN
            BEGIN
                PARSE_QUOTED_NAME ( .FILE_SPEC_LEN,
                    .FILE_SPEC_PTR,
                    DESCRPT );

                ! set a dummy value so it is tested to see if it is VMS spec
                !
                ANSI_NAME_SIZE = 1;
            END
        ELSE
            BEGIN
```



```
319      IF NOT PARSE_NAME_TYPE ( FALSE, ! no wild cards allowed
320      .FILE_SPEC_LEN,
321      .FILE_SPEC_PTR,
322      .FILE_SPEC_PTR,
323      .FILE_SPEC_PTR,
324      THEN ERR_EXIT ( SSS_BADFILENAME );
325      END;
326      END;
327      ! test if the file spec give to us by ATR$ ASCNAME or in quotes is a VMS
328      ! spec so we do not return it in quotes. A size of zero will not work
329      ! cause you need at least a "."
330      IF .ANSI_NAME_SIZE GTR 0
331      THEN
332      BEGIN
333      EXTERNAL WORK_AREA; ! address of general work area
334      DESCRIPT [ 0 ] = FILE_SPEC_MAX;
335      DESCRIPT [ 1 ] = WORK_AREA;
336      IF PARSE_NAME_TYPE ( FALSE, ! no wild cards allowed
337      FILE_SPEC_MAX,
338      FILE_ID[0],
339      DESCRIPT )
340      THEN ANSI_NAME_SIZE = -1;
341      END;
342      ! Fill in the HDR1 FILE ID field and the HDR4 label.
343      CH$MOVE (HD1$$_FILEID, FILE_ID, HDR1[HD1$T_FILEID]);
344      CH$MOVE (HD4$$_FILEID_EXT, FILE_ID[HD1$$_FILEID], HDR4[HD4$T_FILEID_EXT]);
345      ! Check the length of the file name. If the file name will fit in
346      ! the HDR1 FILE ID then set the HDR4 length to zero. Else set up
347      ! the lengths such that the HDR1 FILE ID is filed with the name
348      ! then the remainder of the name is put in the HDR4 label with the
349      ! size that is in the HDR4 label only.
350      ! PLEASE NOTE that the actual implementation of this is different for
351      ! volumes with a 4 in the VOL1 standard field as opposed to a 3 or less.
352      ! This is because the new standard allows us to write any kind
353      ! of data in implementation dependant fields. The old standard did not allow
354      ! us to do this.
355      BEGIN
356      BIND
357      CVT2 = DESCRIPTOR ('!2ZW');
358      LOCAL
359      DESCR : VECTOR [2, LONG],
360      MVL : REF BBLOCK;
361      MVL = .CURRENT_VCB[VCB$L_MVL];
362      IF .FILE_SPEC_LEN LEQU HD1$$_FILEID
363      THEN
364      BEGIN
365      IF .MVL[MVL$B_STDVER] GTR 3
366      THEN
367      HDR4[HD4$B_FILEID_EXT_SIZE] = 0
368      ELSE
369      CH$FILL('0', HD4$$_FILEID_EXT_V3, HDR4[HD4$T_FILEID_EXT_V3]);
370      END
371      END
```

```
ELSE
BEGIN
  IF .MVL[MVLSB_STDVER] GTR 3
  THEN
    HDR4[HD4$B_FILEID_EXT_SIZE] = .FILE_SPEC_LEN - HD1$S_FILEID
  ELSE
  BEGIN
    LOCAL LEN;
    LEN = .FILE_SPEC_LEN - HD1$S_FILEID;
    DESCR[0] = HD4$S_FILEID_EXT_V3;
    DESCR[1] = HDR4[HD4$T_FILEID_EXT_V3];
    $FAO(CVT2,0,DESCR,.LEN);
  END;
END;
END;

! if enter function return file name string to user
!
IF .LOCAL_FIB [ FIB$W_DID_NUM ] NEQ 0
THEN
  RESULTANT_STRING ( .ANSI_NAME_SIZE LSS 0,
                     FILE_SPEC_MAX,
                     FILE_ID[0],
                     .VERSION );

! transform the VMS file version number into ANSI format
!
CALC_TAPE_VER ( .VERSION, GEN_NUM_VER );
DESCRIPT [ 0 ] = HD1$S_GENNO + HDT$S_GENVER;
DESCRIPT [ 1 ] = HDR1 [ HD1$T_GENNO ];
SY$FAO ( DESCRIPTOR ( '!4ZL!2ZL' ), 0, DESCRIPT,
         .GEN_NUM_VER [ 0 ], .GEN_NUM_VER [ 1 ] );

END;
X;

! end of macro FORMAT_FILE_NAME
```

```
413 0793 1 GLOBAL ROUTINE FORMAT_HDRS : COMMON_CALL NOVALUE =
414 0794 1
415 0795 1 ++
416 0796 1
417 0797 1 FUNCTIONAL DESCRIPTION:
418 0798 1 This routine formats HDR1, HDR2, HDR3 and HDR4.
419 0799 1
420 0800 1 CALLING SEQUENCE:
421 0801 1 FORMAT_HDRS()
422 0802 1
423 0803 1 INPUT PARAMETERS:
424 0804 1 none
425 0805 1
426 0806 1 IMPLICIT INPUTS:
427 0807 1 CURRENT_VCB - address of current vcb
428 0808 1 HDR1 - address of HDR1 label
429 0809 1 HDR2 - address of HDR2 label
430 0810 1 HDR3 - address of HDR3 label
431 0811 1 HDR4 - address of HDR4 label
432 0812 1 LOCAL_FIB - copy of user's fib
433 0813 1
434 0814 1 OUTPUT PARAMETERS:
435 0815 1 none
436 0816 1
437 0817 1 IMPLICIT OUTPUTS:
438 0818 1 HDR1, HDR2, HDR3, and HDR4 formatted
439 0819 1
440 0820 1 ROUTINE VALUE:
441 0821 1 none
442 0822 1
443 0823 1 SIDE EFFECTS:
444 0824 1 none
445 0825 1
446 0826 1 --
447 0827 1
448 0828 2 BEGIN
449 0829 2
450 0830 2 EXTERNAL ROUTINE
451 0831 2 CONVDATE_R2J, ! convert regular date to
452 0832 2 ! Julian for tape
453 0833 2 SYSSASCTIM : ADDRESSING_MODE(ABSOLUTE), ! get ASCII date/time
454 0834 2 SYSSFAO : ADDRESSING_MODE(ABSOLUTE), ! format ASCII output
455 0835 2 WRITE_ATTRIBUTE : COMMON_CALL; ! write user supplied attrbts
456 0836 2
457 0837 2
458 0838 2 EXTERNAL REGISTER
459 0839 2 COMMON_REG;
460 0840 2
461 0841 2 EXTERNAL
462 0842 2 CURRENT_UCB : REF BBLOCK, ! address of current UCB
463 0843 2 HDR1 : REF BBLOCK, ! address of HDR1(EOF1) label
464 0844 2 HDR2 : REF BBLOCK, ! address of HDR2(EOF2) label
465 0845 2 HDR3 : REF BBLOCK, ! address of HDR3(EOF3) label
466 0846 2 HDR4 : REF BBLOCK, ! address of HDR3(EOF4) label
467 0847 2 IO_PACKET : REF BBLOCK, ! address of IO request packet
468 0848 2 LOCAL_FIB : BBLOCK; ! copy of user file info block
469 0849 2
```



```

BIND
    CVT4      = DESCRIPTOR ( '14XW' );
    CVT5      = DESCRIPTOR ( '15ZW' );
    DEFAULT   = UPLIT ( '00512' );
    DEF_HEX   = UPLIT ( '0200' );
    STARID    = UPLIT ( 'DECFILE11A' );

GLOBAL
    NMBLOCK   : BBLOCK [10];                ! name block

LOCAL
    DESCR     : VECTOR [2];                ! general Pdescriptor
    MVL       : REF BBLOCK;                ! magtape volume list
    TODAY     : VECTOR [12, BYTE];         ! regular date string

! Blank fill the headers and default the fields
! in HDR1, HDR2, HDR3, and HDR4
CH$FILL( ' ', 320, .HDR1 );
HDR1[HD1$L_HD1LID] = 'HDR1';
MVL = .CURRENT_VCB[VCB$L_MVL];
CH$MOVE( HD1$$FILESETID, MVL[MVL$T_SET_ID], HDR1[HD1$T_FILESETID] );
CH$FILL( '0', HD1$$BLOCKCNT, HDR1[HD1$T_BLOCKCNT] );

! If volume is for interchange do not write any VMS specific fields.
IF NOT .CURRENT_VCB[VCB$V_INTCHG]
    THEN
        CH$MOVE( 10, STARID, HDR1[HD1$T_SYSCODE] );

! default expiration and creation date
DESCR[0] = 12;
DESCR[1] = TODAY;
SY$ASCTIM( 0, DESCR, 0, 0 );                ! get today's date in dd-mmm-yyyy
CONVDATE R2J( TODAY, HDR1[HD1$T_CREATEDT] );
CH$MOVE( HD1$$CREATEDT, HDR1[HD1$T_CREATEDT], HDR1[HD1$T_EXPIREDT] );

!++
! format HDR2 defaults
!--
HDR2[HD2$L_HD2LID] = 'HDR2';
HDR2[HD2$L_RECFORMAT] = 'F';
DESCR[0] = HD2$$BLOCKLEN;
DESCR[1] = HDR2[HD2$T_BLOCKLEN];

! fill in the blocksize
IF NOT $FAO( CVT5, 0, DESCR, .CURRENT_UCB[UCB$W_DEVBUFFSIZ] )
    THEN CH$MOVE( HD2$$BLOCKLEN, DEFAULT, HDR2[HD2$T_BLOCKLEN] );

! default the the record size to be the blocksize
CH$MOVE( HD2$$RECLN, HDR2[HD2$T_BLOCKLEN], HDR2[HD2$T_RECLN] );
```

```
! use the record size if it exists
IF .CURRENT_VCB[VCB$W_RECORDSZ] NEQ 0
THEN
  BEGIN
    DESCR[0] = HD2$S_RECLEN;
    DESCR[1] = HDR2[HD2$T_RECLEN];

    IF NOT $FAO(CVT5, 0, DESCR, .CURRENT_VCB[VCB$W_RECORDSZ])
    THEN
      CH$MOVE(HD2$S_RECLEN, HDR2[HD2$T_BLOCKLEN], HDR2[HD2$T_RECLEN]);
    END;

    HDR2[HD2$T_BUFOFF] = '00';

    ++
    default HDR3 ( sequential files, fixed length block size )
    --
    HDR3[HD3$S_HD3LID] = 'HDR3';
    CH$FILL('0', HD3$S_RECATR, HDR3[HD3$T_RECATR]);
    (HDR3[HD3$T_RECATR] + 4) < 0, 32 > = '0201';

    ++
    default HDR4 no long file name, make the default dependant on the
    ANSI version type
    --
    HDR4[HD4$S_HD4LID] = 'HDR4';
    IF .MVL[MV[$B_STDVER] GTR 3
    THEN
      HDR4[HD4$B_FILEID_EXT_SIZE] = 0
    ELSE
      CH$FILL('0', HD4$S_FILEID_EXT_V3, HDR4[HD4$T_FILEID_EXT_V3]);

    ++
    fill in the RMS default record size
    if record size on mount then use it
    else if blocks size the use it
    else default
    --
    DESCR[0] = 4;
    DESCR[1] = HDR3[HD3$T_RECATR];
    IF .CURRENT_VCB[VCB$W_RECORDSZ] NEQ 0
    THEN
      BEGIN
        IF NOT $FAO( CVT4, 0, DESCR, .CURRENT_VCB[VCB$W_RECORDSZ] )
        THEN
          BEGIN
            IF NOT $FAO( CVT4, 0, DESCR, .CURRENT_UCB[UCB$W_DEVBUFSIZ] )
            THEN CH$MOVE ( 4, DEF_HEX, HDR3[HD3$T_RECATR] );
          END;
        END;
      END;
```

```
0964 END
0965 ELSE
0966 BEGIN
0967 IF NOT $FAO( CVT4, 0, DESCR, .CURRENT_UCB[UCBSW_DEVBUFFSIZ] )
0968 THEN CH$MOVE ( 4, DEF_HEX, HDR3[HDR3$T_RECATR] );
0969 END;

! pickup user supplied attributes
0970 CH$FILL ( 0, 10, NMBLOCK );
0971 WRITE_ATTRIBUTE ();

! +
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
1
END;

! end of routine FORMAT_HDRS

.TITLE FRMHDR
.IDENT \V04-000\

.PSECT $CODE$,NOWRT,2

57 58 34 21 00000 P.AAB: .ASCII \!4XW\
00000004 00004 P.AAA: .LONG 4
00000000 00008 .ADDRESS P.AAB
57 5A 35 21 0000C P.AAD: .ASCII \!5ZW\
00000004 00010 P.AAC: .LONG 4
00000000 00014 .ADDRESS P.AAD
00 00 00 32 31 35 30 30 00018 P.AAE: .ASCII \00512\<0><0><0>
30 30 32 30 00020 P.AAF: .ASCII \0200\
00 00 41 31 31 45 4C 49 46 43 45 44 00024 P.AAG: .ASCII \DECF1E11A\<0><0>
57 5A 32 21 00030 P.AAI: .ASCII \!2ZW\
00000004 00034 P.AAH: .LONG 4
00000000 00038 .ADDRESS P.AAI
4C 5A 32 21 4C 5A 34 21 0003C P.AAK: .ASCII \!4ZL!2ZL\
00000008 00044 P.AAJ: .LONG 8
00000000 00048 .ADDRESS P.AAK

.PSECT $LOCKEDD1$,NOEXE,2

00000 NMBLOCK::
.BKLB 10

CVT4= P.AAA
CVT5= P.AAC
DEFAULT= P.AAE
DEF_HEX= P.AAF
STARID= P.AAG
CVT2= P.AAH

.EXTRN CONVDAT_R2J, SYSSASCTIM
.EXTRN SYSSFAO, WRITE_ATTRIBUTE
```


.EXTRN CURRENT_UCB, HDR1
.EXTRN HDR2, HDR3, HDR4
.EXTRN IO_PACKET, LOCAL_FIB
.EXTRN CALC_TAPE_VER, PARSE_NAME_TYPE
.EXTRN PARSE_QUOTED_NAME
.EXTRN RESULTANT_STRING
.EXTRN STRIP_VERSION, ANSI_NAME_SIZE
.EXTRN WORK_AREA

.PSECT \$CODE\$,NOWRT,2

.ENTRY FORMAT_HDRS, Save R2,R3,R4,R5,R6,R7,R8,R9,- R10 0793
MOVAB SYS\$FAO, R10
MOVAB -220(SP), SP
MOVL HDR1, R6 0869
MOVCS #0, (SP), #32, #320, (R6)

MOVL #827475016, (R6) 0870
MOVL 52(CURRENT_VCB), MVL 0871
MOVCS #6, 12(MVL), 21(R6) 0872
MOVCS #0, (SP), #48, #6, 54(R6) 0873

BBS #4, 44(CURRENT_VCB), 1\$ 0877
MOVCS #10, STARID, 60(R6) 0879
MOVL #12, DESCR 0883
MOVAB TODAY, DESCR+4 0884
CLRQ -(SP) 0885
PUSHAB DESCR
CLRL -(SP)
CALLS #4, @SYS\$ASCTIM
ADJL3 #41, HDR1, -(SP) 0886
PUSHAB TODAY
CALLS #2, CONVDAT_R2J
MOVL HDR1, R0 0887
MOVCS #6, 41(R0), 47(R0)
MOVL HDR2, R0 0894
MOVL #844252232, (R0)
MOVB #70, 4(R0) 0895
MOVL #5, DESCR 0896
MOVAB 5(R0), DESCR+4 0897
MOVL CURRENT_UCB, R0 0901
MOVZWL 66(R0), -(SP)
PUSHAB DESCR
CLRL -(SP)
PUSHAB CVT5
CALLS #4, SYS\$FAO
BLBS R0, 2\$
MOVL HDR2, R0 0902
MOVCS #5, DEFAULT, 5(R0)
MOVL HDR2, R6 0906
MOVCS #5, 5(R6), 10(R6)
MOVZWL 80(CURRENT_VCB), R8 0910
CLRL R9
TSTL R8
BEQL 3\$
INCL R9

0140 8F 20 5A 00000000G 00 9E 00002
5E FF24 CE 9E 00009
56 0000G CF D0 0000E
6E 00 2C 00013
66 0001A
31524448 8F D0 0001B
57 34 AB D0 00022
15 A6 0C A7 06 28 00026
06 30 6E 00 2C 0002C
36 A6 00031
06 2C AB 04 E0 00033
3C A6 9C AF 0A 28 00038
F8 AD 0C D0 0003E 1\$:
FC AD EC AD 9E 00042
F8 AD 7E 7C 00047
7E D4 00049
00000000G 9F 04 FB 0004E
7E 0000G CF 29 C1 00055
EC AD 9F 0005B
0000G CF 02 FB 0005E
50 0000G CF D0 00063
2F A0 29 A0 06 28 00068
50 0000G CF D0 0006E
60 32524448 8F D0 00073
04 A0 46 8F 90 0007A
F8 AD 05 D0 0007F
FC AD 05 A0 9E 00083
50 0000G CF D0 00088
7E 42 A0 3C 0008D
F8 AD 9F 00091
FF2A 7E D4 00094
CF 9F 00096
6A 04 FB 0009A
0C 50 E8 0009D
50 0000G CF D0 000A0
05 A0 FF22 CF 05 28 000A5
56 0000G CF D0 000AC 2\$:
0A A6 05 A6 05 28 000B1
58 50 AB 3C 000B7
59 D4 000BB
58 D5 000BD
27 13 000BF
59 D6 000C1

51	50	58	11	0019D	BRB	17\$
51	52	54	C1	0019F	ADDL3	K, 1, R1
51	52	41	3C	001A3	MOVZWL	NMBLOCK[R1], R2
51	52	8F	C7	001A9	DIVL3	#1600, R2, R1
51	52	51	90	001B1	MOVB	R1, CHARS
51	52	28	C7	001B4	DIVL3	#40, R2, R1
51	51	01	7A	001B8	EMUL	#1, R1, #0, -(SP)
51	51	28	7B	001BD	EDIV	#40, (SP)+, R1, R1
01	AE	51	90	001C2	MOVB	R1, CHARS+1
00	52	01	7A	001C6	EMUL	#1, R2, #0, -(SP)
51	52	28	7B	001CB	EDIV	#40, (SP)+, R1, R1
02	AE	51	90	001D0	MOVB	R1, CHARS+2
		51	D4	001D4	CLRL	J
53		6E	9A	001D6	MOVZBL	CHARS[J], R3
		1A	13	001DA	BEQL	16\$
52		AE	9E	001DC	MOVAB	NAME_STRING, R2
1E		53	91	001E0	CMPB	R3, #30
		06	1E	001E3	BGEQU	14\$
53		A3	9E	001E5	MOVAB	64(R3), R3
		03	11	001E9	BRB	15\$
53		12	C0	001EB	ADDL2	#18, R3
OC BE42		53	90	001EE	MOVAB	R3, @FILE_SPEC_LEN[R2]
		AE	D6	001F3	INCL	FILE_SPEC_LEN
DC		02	F3	001F6	AOBLEQ	#2, J, 13\$
A1		55	F3	001FA	AOBLEQ	R5, 1, 12\$
		54	D5	001FE	TSTL	K
		OC	12	00200	BNEQ	18\$
		AE	9E	00202	MOVAB	NAME_STRING, R0
OC BE40		2E	90	00206	MOVAB	#46, @FILE_SPEC_LEN[R0]
		AE	D6	0020B	INCL	FILE_SPEC_LEN
		03	F1	0020E	ACBL	#3, #3, K, 9\$
54		CF	B0	00214	MOVW	NMBLOCK+8, VERSION
		AE	9E	00219	MOVAB	NAME_STRING, FILE_SPEC_PTR
08		01	8A	0021E	BICB2	#1, QUOTED_NAME
04		25	11	00222	BRB	20\$
		A0	3C	00224	MOVZWL	18(ABD), FILE_SPEC_LEN
OC		A0	9E	00229	MOVAB	16(ABD), R1
		61	3C	0022D	MOVZWL	(R1), R0
08		A1	9E	00230	MOVAB	1(R1)[R0], FILE_SPEC_PTR
		AE	9F	00236	PUSHAB	QUOTED_NAME
		7E	D4	00239	CLRL	-(SP)
		AE	9F	0023B	PUSHAB	FILE_SPEC_PTR
		AE	9F	0023E	PUSHAB	FILE_SPEC_LEN
0000G	CF	04	FB	00241	CALLS	#4, STRIP-VERSION
7FFF	8F	50	B0	00246	MOVW	R0, VERSION
		57	B1	00249	CMPW	VERSION, #32767
		04	1B	0024E	BLEQU	21\$
004F	8F	8F	BF	00250	CHMU	#2080
		00	2C	00254	MOVCS	#0, (SP), #32, #79, FILE_ID
20	6E	AE		0025B		
		CF	95	0025D	TSTB	ANSI_NAME_SIZE
		3A	18	00261	BGEQ	23\$
E4	AD	8F	9A	00263	MOVZBL	#79, DESCRIPT
E8	AD	AE	9E	00268	MOVAB	FILE_ID, DESCRIPT+4
	15	AE	E9	0026D	BLBC	QUOTED_NAME, 22\$
		AD	9F	00271	PUSHAB	DESCRIPT
		OC	DD	00274	PUSHL	FILE_SPEC_PTR

			14	AE	DD	00277	PUSHL	FILE_SPEC_LEN	
	0000G	CF		03	FB	0027A	CALLS	#3, PARSE_QUOTED_NAME	
	0000G	CF		01	90	0027F	MOVB	#1, ANSI_NAME_SIZE	
				17	11	00284	BRB	23\$	
			E4	AD	9F	00286	PUSHAB	DESCRIPT	
			OC	AE	DD	00289	PUSHL	FILE_SPEC_PTR	
			14	AE	DD	0028C	PUSHL	FILE_SPEC_LEN	
				7E	D4	0028F	CLRL	-(SP)	
	0000G	CF		04	FB	00291	CALLS	#4, PARSE_NAME_TYPE	
		04		50	E8	00296	BLBS	R0, 23\$	
			0818	8F	BF	00299	CHMU	#2072	
			0000G	CF	95	0029D	TSTB	ANSI_NAME_SIZE	
				24	15	002A1	BLEQ	24\$	
	E4	AD		4F	8F	9A	MOVZBL	#79, DESCRIPT	
	E8	AD	0000G	CF	9E	002AB	MOVAB	WORK AREA, DESCRIPT+4	
				E4	AD	9F	PUSHAB	DESCRIPT	
				24	AE	9F	PUSHAB	FILE_ID	
				4F	8F	9A	MOVZBL	#79, -(SP)	
				7E	D4	002B8	CLRL	-(SP)	
	0000G	CF		04	FB	002BA	CALLS	#4, PARSE_NAME_TYPE	
		05		50	E9	002BF	BLBC	R0, 24\$	
	0000G	CF		01	8E	002C2	MNEGB	#1, ANSI_NAME_SIZE	
04	A0	20	0000G	CF	D0	002C7	MOVL	HDR1, R0	
				11	28	002CC	MOV3	#17, FILE_ID, 4(R0)	
05	A6	31	0000G	CF	D0	002D2	MOVL	HDR4, R6	
				3E	28	002D7	MOV3	#62, FILE_ID+17, 5(R6)	
				34	AB	D0	MOVL	52(CURRENT VCB), MVL	
				OC	AE	D1	CMPL	FILE_SPEC_LEN, #17	
				13	1A	002E5	BGTRU	26\$	
				03	A0	91	CMPB	34(MVL), #3	
				05	1B	002E7	BLEQU	25\$	
				04	A6	94	CLRB	4(R6)	
				33	11	002F0	BRB	28\$	
	43	A6	3030	8F	B0	002F2	MOVW	#12336, 67(R6)	
				2B	11	002F8	BRB	28\$	
51	OC	AE		11	C3	002FA	SUBL3	#17, FILE_SPEC_LEN, R1	
		03		A0	91	002FF	CMPB	34(MVL), #3	
				06	1B	00303	BLEQU	27\$	
	04	A6		51	90	00305	MOVB	R1, 4(R6)	
				1A	11	00309	BRB	28\$	
				51	D0	0030B	MOVL	R1, LEN	
		50		02	D0	0030E	MOVL	#2, DESCR	
	10	AE		43	A6	9E	MOVAB	67(R6), DESCR+4	
	14	AE		50	DD	00317	PUSHL	LEN	
				14	AE	9F	PUSHAB	DESCR	
				7E	D4	0031C	CLRL	-(SP)	
			FCC6	CF	9F	0031E	PUSHAB	CVT2	
				04	FB	00322	CALLS	#4, SYSSFAO	
			0000G	CF	B5	00325	TSTW	LOCAL_FIB+10	
				19	13	00329	BEQL	30\$	
				57	3C	0032B	MOVZWL	VERSION, -(SP)	
				24	AE	9F	PUSHAB	FILE_ID	
				7E	4F	8F	MOVZBL	#79, -(SP)	
				7E	D4	00335	CLRL	-(SP)	
			0000G	CF	95	00337	TSTB	ANSI_NAME_SIZE	
				02	18	0033B	BGEQ	29\$	
				6E	D6	0033D	INCL	(SP)	

```
0000G CF      18 04 FB 0033F 29$: CALLS #4, RESULTANT_STRING
                        AE 9F 00344 30$: PUSHAB GEN_NUM_VER
                        57 3C 00347 MOVZWL VERSION, -(SP)
0000G CF      02 FB 0034A CALLS #2, CALC_TAPE_VER
    E4 AD      06 D0 0034F MOVL #6, DESCRIPT
0000G CF      23 C1 00353 ADDL3 #35, HDR1, DESCRIPT+4
    E8 AD      1C AE DD 0035A PUSHL GEN_NUM_VER+4
                        1C AE DD 0035D PUSHL GEN_NUM_VER
    E4 AD      7E D4 00360 PUSHAB DESCRIPT
                        7E D4 00363 CLRL -(SP)
                        CF 9F 00365 PUSHAB P.AAJ
    6A FC8F      05 FB 00369 CALLS #5, SYSSFAO
                        04 0036C RET
```

0984

; Routine Size: 877 bytes, Routine Base: \$CODE\$ + 004C

```
: 605      0985 1
: 606      0986 1 END
: 607      0987 1
: 608      0988 0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	953	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
\$LOCKEDD1\$	10	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	50	0	1000	00:01.9

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:FRMHDR/OBJ=OBJ\$:FRMHDR MSRC\$:FRMHDR/UPDATE=(ENH\$:FRMHDR)

```
: 609      0989 0
: Size:      877 code + 86 data bytes
: Run Time:   00:19.9
```

FRMHDR
V04-000

E 13
16-Sep-1984 02:19:38

VAX-11 Bliss-32 V4.0-742

Page 18

; Elapsed Time: 00:38.7
; Lines/CPU Min: 2989
; Lexemes/CPU-Min: 23271
; Memory Used: 299 pages
; Compilation Complete

FRM
V04

0254 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY